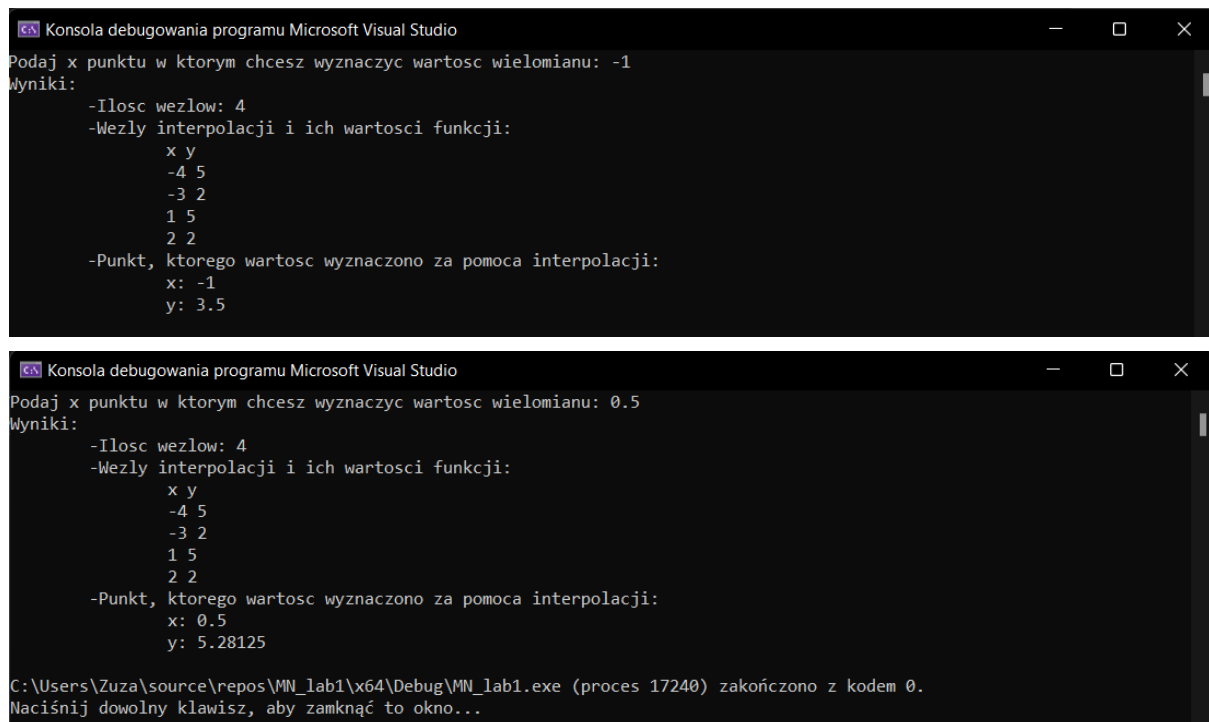


Zadanie 1: wyznaczenie wartości wielomianu interpolacyjnego Lagrange'a w danym punkcie

Do obliczenia wartości wielomianu, posłużono się wzorami podanymi na zajęciach, w miejsce x wstawiając punkt, dla którego poszukujemy wartości. Kod programu obliczającego szukaną wartość zaprezentowano poniżej:

```
1  #include <iostream>
2  #include <vector>
3  #include <fstream>
4  #include <string>
5  #include <stdio.h>
6
7  using namespace std;
8
9  int main()
10 {
11     //deklaracje zmiennych
12     vector<pair<double, double>> wezly; //wektor na wezly interpolacji, porzedstawione jako pary doubli, first = x, second = y
13     vector<double> wspolczynniki; //wektor na skladniki wielomianu lagrange'a
14     int n; //ilosc wezlow
15     pair<double, double> punkt; //punkt w ktorym obliczamy wartosci wezlow
16
17     //pobieranie danych wejsciowych
18     ifstream czytaj("dane.txt"); //wczytywanie z pliku
19     czytaj >> n; //wczytanie ilosci wezlow
20     for (int i = 0; i < n; ++i) //wczytywanie wspolrzednych wezlow
21     {
22         double a, b;
23         czytaj >> a >> b;
24         pair<double, double> pomoc = make_pair(a, b);
25         wezly.push_back(pomoc);
26     }
27     cout << "Podaj x punktu w ktorym chcesz wyznaczyc wartosc wielomianu: ";
28     cin >> punkt.first; //podanie x dla ktorego y chcemy interpolowac
29     punkt.second = 0; //interpolowany y na razie jest rowny 0
30
31     //obliczanie wartosci wielomianu lagrange'a
32     //krok 1 - obliczanie iloczynow dla kazdego wezla z podstawieniem wspolrzednych szukanego punktu w miejsce x:
33     for (int i = 0; i < n; ++i)
34     {
35         double skladnik = 1.0; //na poczatku skladniki wielomianu lagrange'a sa rowne 1
36         for (int j = 0; j < n; ++j) //petla obliczajaca iloczyny
37         {
38             if (i == j)
39             {
40                 continue;
41             }
42             else
43             {
44                 skladnik *= (punkt.first - wezly[j].first) / (wezly[i].first - wezly[j].first); //(x - xj)/(xi - xj)
45             }
46         }
47         wspolczynniki.push_back(skladnik);
48     }
49
50     //krok 2 - przemnozenie wspolczynnika razy wartosci w wezlach i dodanie ich do siebie
51     for (int i = 0; i < n; ++i) //petla sumujaca
52     {
53         punkt.second += wezly[i].second * wspolczynniki[i]; //fx * (x - xj)/(xi - xj)
54     }
55
56     //wypisanie wynikow
57     cout << "Wyniki:\n\t-Ilosc wezlow: " << n << "\n\t-Wezly interpolacji i ich wartosci funkcji: \n\t\tx y\n";
58     for (auto i : wezly)
59     {
60         cout << "\t\t" << i.first << " " << i.second << "\n";
61     }
62     cout << "\t-Punkt, ktorego wartosc wyznaczono za pomoca interpolacji: \n\t\tx: " << punkt.first << "\n\t\t y: " << punkt.second << "\n";
63     return 0;
64 }
```

W liniach 33 - 48 wyznaczano składniki wyniku dla każdego węzła interpolacji. W liniach 50 - 54 każdy z policzonych wcześniej składników pomnożono przez wartość funkcji w odpowiadającym węźle interpolacji i dodano do wyniku, który następnie wyświetlono na ekran. W celu sprawdzenia poprawności działania wykonano sprawdzenie dla punktów -1 i 0,5. Wyniki zaprezentowano na poniższych zrzutach ekranu:



```
Konsola debugowania programu Microsoft Visual Studio
Podaj x punktu w którym chcesz wyznaczyć wartość wielomianu: -1
Wyniki:
-Ilość węzłów: 4
-Wezły interpolacji i ich wartości funkcji:
  x y
  -4 5
  -3 2
  1 5
  2 2
-Punkt, którego wartość wyznaczono za pomocą interpolacji:
  x: -1
  y: 3.5

Konsola debugowania programu Microsoft Visual Studio
Podaj x punktu w którym chcesz wyznaczyć wartość wielomianu: 0.5
Wyniki:
-Ilość węzłów: 4
-Wezły interpolacji i ich wartości funkcji:
  x y
  -4 5
  -3 2
  1 5
  2 2
-Punkt, którego wartość wyznaczono za pomocą interpolacji:
  x: 0.5
  y: 5.28125

C:\Users\Zuza\source\repos\MN_lab1\x64\Debug\MN_lab1.exe (proces 17240) zakończono z kodem 0.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Zadanie 2: wyznaczenie wartości $\sqrt[3]{50}$ za pomocą interpolacji

Aby obliczyć wartość $\sqrt[3]{50}$, wykorzystano własności funkcji $\sqrt[3]{x}$. Dzięki temu wyznaczono 4 punkty interpolacji:

lp	x	y
1.	27	3
2.	64	4
3.	125	5
4.	216	6

Następnie zastosowano interpolację wielomianową Lagrange'a w taki sam sposób, jak w poprzednim zadaniu. Wykorzystano wzory podane na zajęciach, w miejsce x podstawiając 50. Kod programu pokazano poniżej:

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main()
7  {
8      //deklaracja zmiennych
9      vector <pair <double, double>> wezly; //wspolrzedne wezlow interpolacji w postaci par x, y
10     vector <double> wspolczynniki; //do obliczania wspolczynnika wielomianu lagrange'a
11     int n = 4; //ilosc wezlow interpolacji
12     pair <double, double> punkt;
13     punkt.first = 50.0;
14     punkt.second = 0.0;
15     pair <double, double> pomoc; //do tworzenia wezlow interpolacji
16
17     //generowanie danych
18     for (int i = 0; i < n; ++i)
19     {
20         double y = 3.0 + i;
21         double x = y * y * y;
22         pomoc.first = x;
23         pomoc.second = y;
24         wezly.push_back(pomoc);
25     }
26     //obliczanie wartosci wielomianu lagrange'a
27     //krok 1 - obliczanie iloczynow dla kazdego wezla z podstawieniem wspolrzednych szukanego punktu w miejsce x:
28     for (int i = 0; i < n; ++i)
29     {
30         double skladnik = 1.0; //na poczatku skladniki wielomianu lagrange'a sa rowne 1
31         for (int j = 0; j < n; ++j) //petla obliczajaca iloczyny
32         {
33             if (i == j)
34             {
35                 continue;
36
37             else
38             {
39                 skladnik *= (punkt.first - wezly[j].first) / (wezly[i].first - wezly[j].first); //(x - xj)/(xi - xj)
40             }
41         }
42         wspolczynniki.push_back(skladnik);
43     }
44
45     //krok 2 - przemnozenie wspolczynnika razy wartosci w wezlach i dodanie ich do siebie
46     for (int i = 0; i < n; ++i) //petla sumujaca
47     {
48         punkt.second += wezly[i].second * wspolczynniki[i]; //fx * (x - xj)/(xi - xj)
49     }
50     //sprawdzenie poprawnosci generowanych danych i wynikow
51     cout << "Wyniki:\n\t-Ilosc wezlow: " << n << "\n\t-Wezly interpolacji i ich wartosci funkcji: \n\t\tx y\n";
52     for (auto i : wezly)
53     {
54         cout << "\t\t" << i.first << " " << i.second << "\n";
55     }
56     cout << "\t-Punkt, ktorego wartosc wyznaczono za pomoca interpolacji: \n\t\tx: " << punkt.first << "\n\t\tty: " << punkt.second << "\n";
57     return 0;
58 }

```

Uzyskano następujące wyniki:

```

Konsola debugowania programu Microsoft Visual Studio
Wyniki:
-Ilosc wezlow: 4
-Wezly interpolacji i ich wartosci funkcji:
  x y
  27 3
  64 4
  125 5
  216 6
-Punkt, ktorego wartosc wyznaczono za pomoca interpolacji:
  x: 50
  y: 3.66588

C:\Users\Zuza\source\repos\MN_lab1\x64\Debug\MN_lab1.exe (proces 17704) zakończono z kodem 0.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

Wnioski:

Dzięki zaimplementowaniu interpolacji wielomianowej Lagrange'a można wyznaczyć wartości funkcji znając jej postać (tak jak w zadaniu 2), lub posiadając tylko informację o punktach przez które ona przechodzi. Niestety dokładność obliczeń nie zawsze jest zadowalająca - w zadaniu 2 otrzymano 3,66588 gdy w rzeczywistości wynik jest bliższy 3,68.